

5.455 / H / 02



PEMBUATAN ALAT UNTUK MENGUKUR ARAH ANGIN, KECEPATAN ANGIN DAN SUHU UDARA

TUGAS AKHIR



Oleh :

A. SYAMSUL ARIFIN
NRP. 2296030084

SAM HADI HIDAYAT

RSE
681.2
Ari
P-1
2000

PEMBUATAN ALAT UNTUK MENGUKUR ARAH ANGIN, KECEPATAN ANGIN DAN SUHU UDARA

TUGAS AKHIR

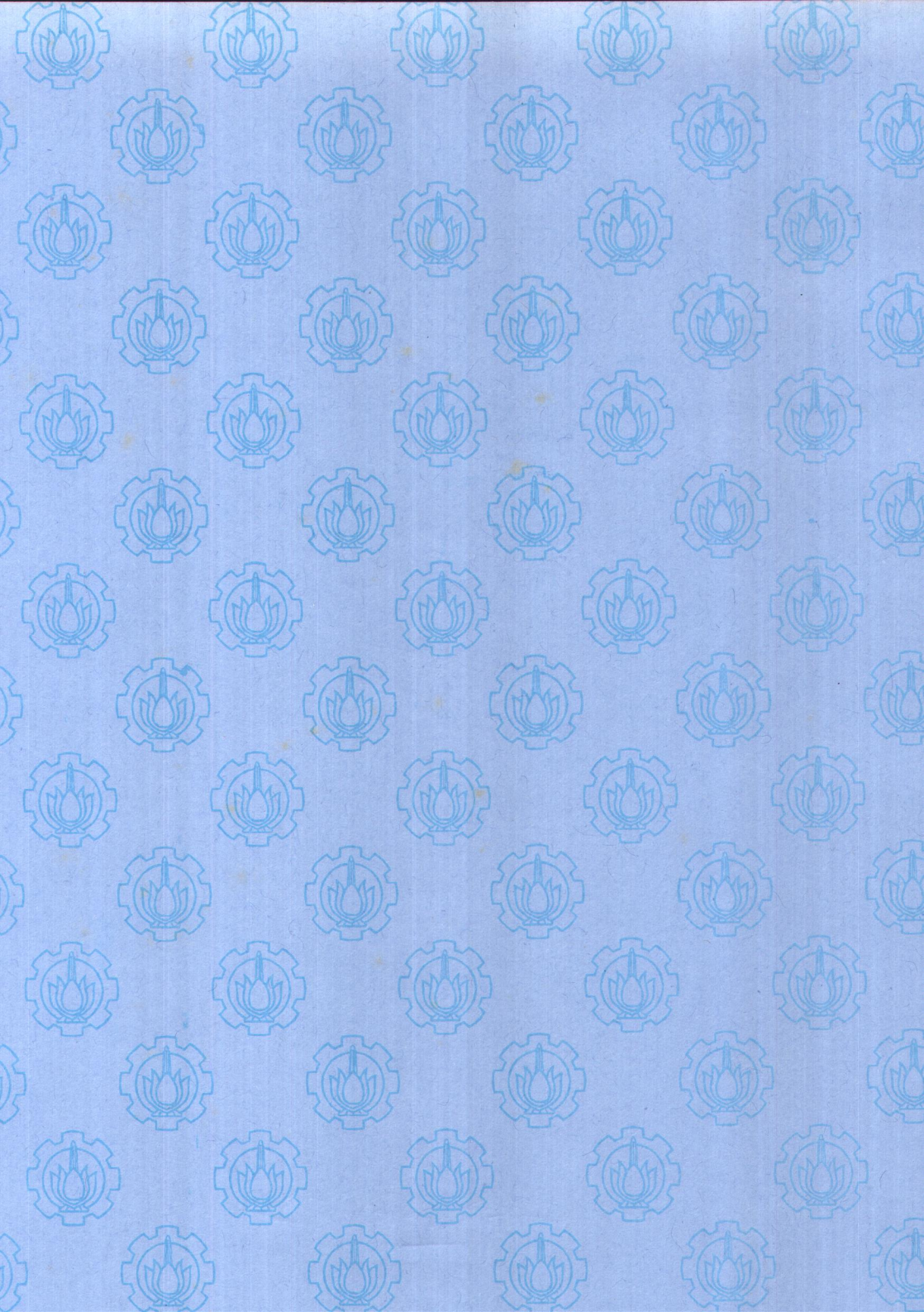
**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Menyelesaikan Studi Program Diploma III Komputer Kontrol
Pada
Jurusan Teknik Elektro
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya**

**Mengetahui / Menyetujui
Dosen Pembimbing**



Ir. RUSDHIYANTO EAK, M.T.

NIP. 121.435.676



ABSTRAK

Informasi tentang arah angin, kecepatan angin dan suhu udara, sangat penting dalam penerbangan misalnya bandara-bandara. Karena angin merupakan faktor utama dalam penerbangan pesawat terbang, jika tidak ditunjang dengan pengukuran yang tepat akan berakibat fatal.

Oleh karena itu kami merancang dan membuat alat untuk mengukur arah angin, kecepatan angin dan suhu udara. Prinsip kerja alat ini adalah motor stepper sebagai pembawa optocoupler yang merupakan sensor untuk mencari lobang yang ada pada piringan. Untuk mengukur arah angin juga menggunakan optocoupler dengan cara perhitungan jumlah pulsa yang diterima, dan untuk mengukur suhu udara kami menggunakan LM35 yang dibantu dengan mikrokontroler 80486. Semua komponen atau rangkaian diatas dihubungkan dengan IBM PC melalui serial port dan Delphi sebagai bahasa pemrograman.

KATA PENGANTAR

Puji syukur kehadirat ALLAH SWT atas berkat dan rahmatnya, sehingga penulis menyelesaikan Tugas Akhir yang berjudul :

“PEMBUATAN ALAT UNTUK MENGUKUR ARAH ANGIN, KECEPATAN ANGIN DAN SUHU UDARA”

Tugas Akhir ini merupakan salah satu syarat yang harus ditempuh untuk menyelesaikan studi pada Program Studi DIII Komputer Kontrol, Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember.

Dalam Tugas Akhir ini, penulis banyak mendapat bantuan dan bimbingan berbagai pihak. Didalam kesempatan ini penulis menyampaikan terima kasih kepada :

1. Ir. H.M. Djoko Santoso, selaku Ketua Program Studi DIII Komputer Kontrol, Jurusan Teknik Elektro, Fakultas Teknologi Industri Institut Teknologi Sepuluh Nopember.

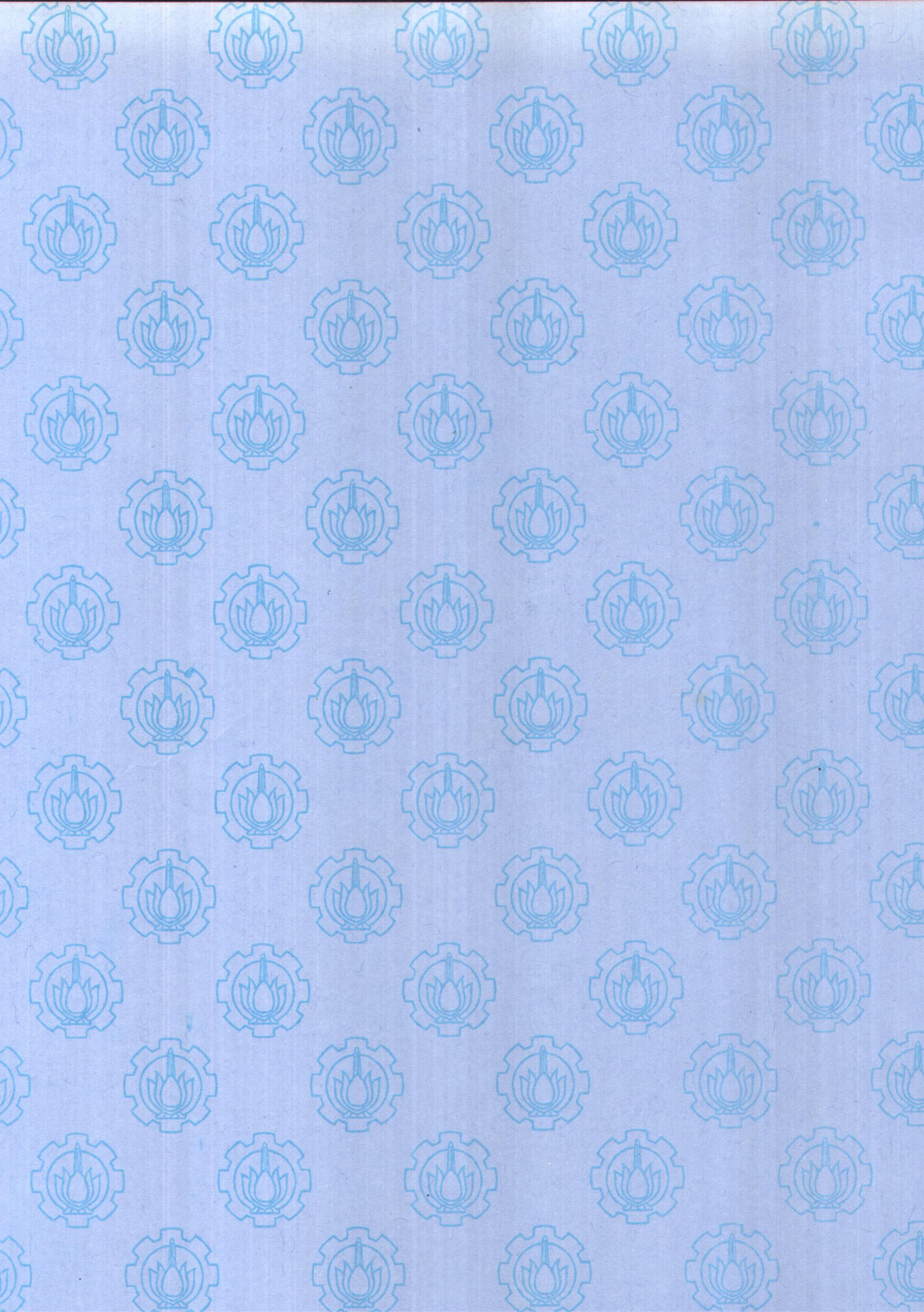
2. Ir. Budhiyanto, EAK, MT, selaku Dosen Pembimbing tugas akhir yang telah

5. Seluruh staf Dosen Pengajar yang memberikan bimbingan selama perkuliahan.
6. Ibunda dan Ayahanda tercinta yang selalu memberikan dorongan baik secara moril maupun materil pada kami.
7. Teman-teman MANLABS CREW (Titus, Ali, Nanang, Toin, Totok, Taufik, Yudi, Paidi, Mr. Broods, Coco, Agos, Eko, Imamul, Anang, dll) Terima kasih atas bantuan kalian semua semoga sukses selalu.

Akhir kata penulis berharap semoga hasil dari tugas akhir ini dapat bermanfaat untuk an ilmu pengetahuan dan teknologi.

Surabaya, Nopember 1999

Penulis



DAFTAR ISI

	HALAMAN
SAHAN	
AK	i
PENGANTAR	ii
AR ISI	iv
AR GAMBAR	vi
AR TABEL	vii
: PENDAHULUAN	
I.1. LATAR BELAKANG	1
I.2. PERMASALAHAN	1
I.3. PEMBATAKAN MASALAH	1
I.4. MAKSUD DAN TUJUAN	2
I.5. SISTEMATIKA PEMBAHASAN	2
: TEORI PENUNJANG	
II.1.1. Programmable Peripheral Interface 8255 (PPI 8255A).....	3
II.1.2. Pengaturan Mode PPI 8255	5

III.2.1. Rangkaian ADC	15
III.2.2. Rangkaian Driver Motor Stepper	16
III.2.3. Rangkaian Optocoupler	17
III.2.4. Flowchart	18

V : PENGUJIAN ALAT

IV.1. Pengujian PPI 8255	19
IV.2. Pengujian Driver Motor Stepper	20
IV.3. Pengujian Rangkaian Optocoupler	22
IV.4. Pengujian Sensor Suhu LM35	23

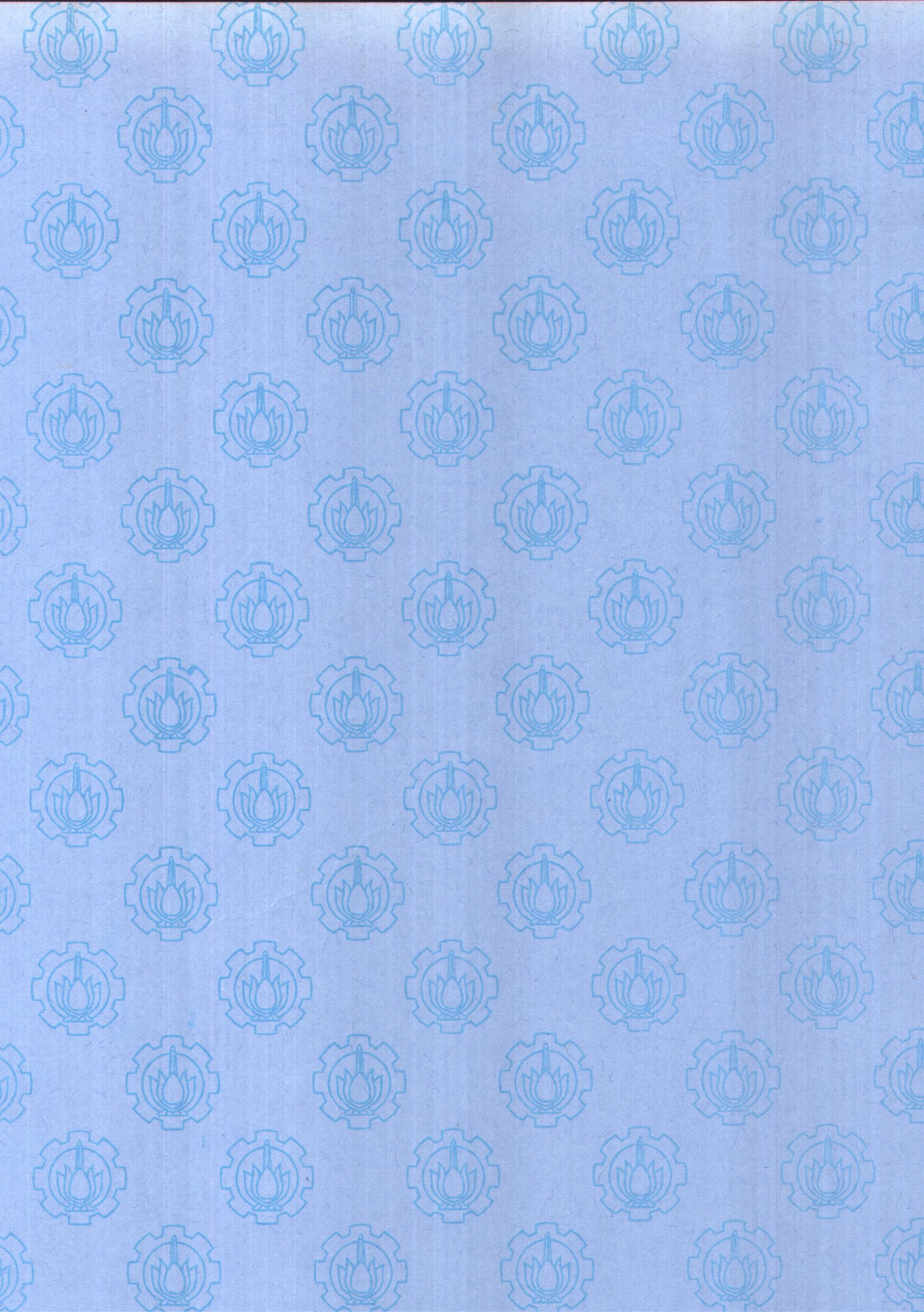
: PENUTUP

V.1. Kesimpulan	25
V.2. Saran	26

DAFTAR PUSTAKA

DAFTAR

KESEHATAN HIDUP

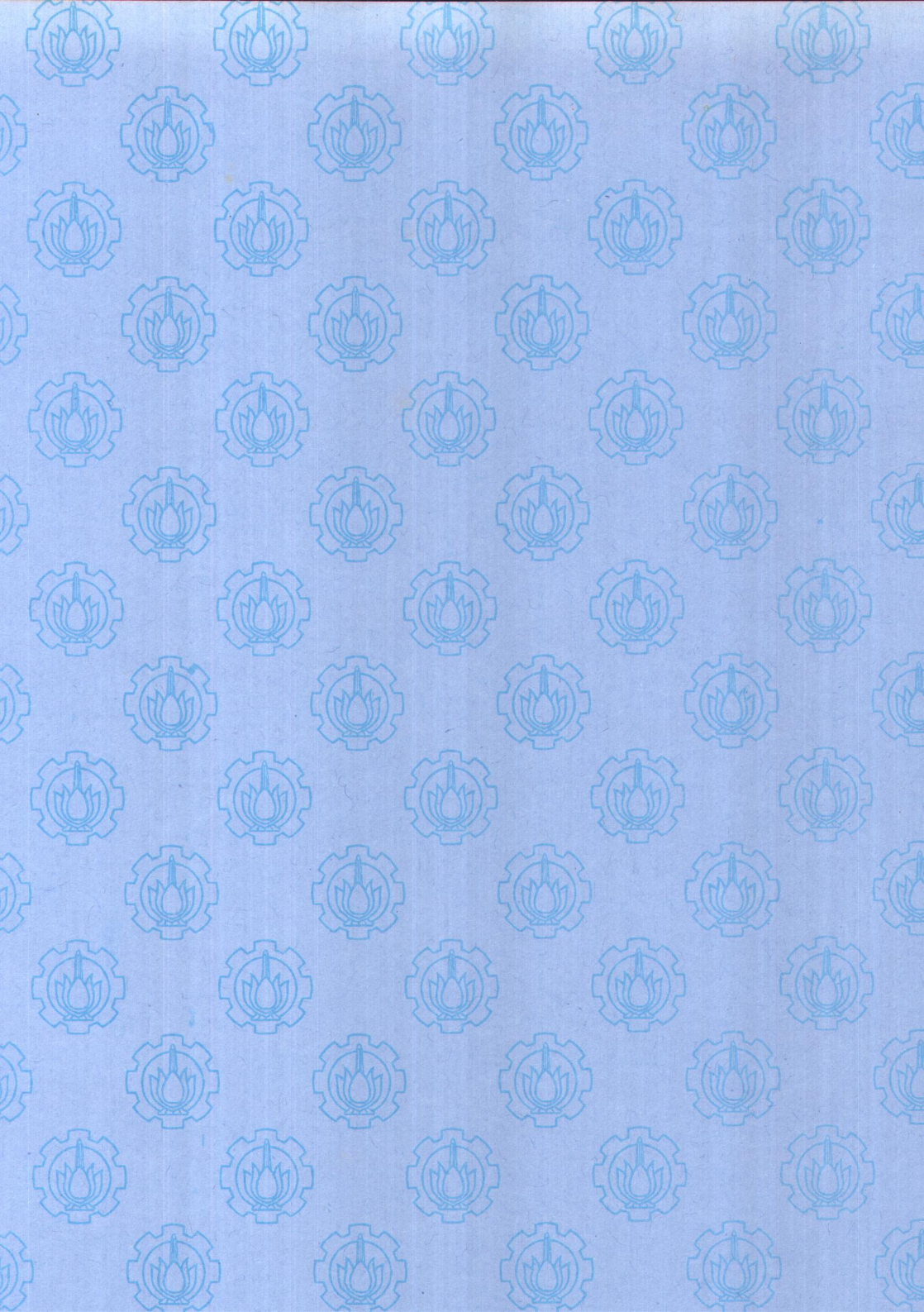


DAFTAR GAMBAR

	HALAMAN
r 2.1 Blok Diagram PPI 8255	5
r 2.2 Control Word PPI 8255	7
r 2.3 Konfigurasi Pin ADC 0804	9
r 2.4 Prinsip Kerja Motor Stepper	11
r 2.5 Fisik Optocoupler dan Isi Rangkaiannya	13
r 3.1 Blok Diagram Alat	14
r 3.2 Bentuk Fisik Alat Pengukur Arah Angin	14
r 3.3 Bentuk Fisik Alat Pengukur Kecepatan Angin	15
r 3.4 Rangkaian ADC	16
r 3.5 Rangkaian Driver Motor Stepper	17
r 3.6 Rangkaian Optocoupler	17
r 3.7 Flowchart	18
r 4.1 Pengujian Rangkaian Driver Motor Stepper	22
r 4.2 Pengujian Rangkaian Optocoupler	23
r 4.3 Cara Pengukuran LM35	23

DAFTAR TABEL

	HALAMAN
1. Operasi Dasar PPI 8255	6
2. Mode Pengoperasian PPI 8255	7
3. (a) Logika Untuk Putaran Full Step	11
3. (b) Logika Untuk Putaran Half Step	12
1. Hasil Keluaran Logika “1” PPI 8255	20
2. Hasil Keluaran Logika “0” PPI 8255	20
3. Hasil Pengukuran Rangkaian Optocoupler	22
4. Data Pengukuran LM35	24



BAB I

PENDAHULUAN

LATAR BELAKANG

Informasi sangat penting bagi kehidupan kita, karena dengan informasi kita dapat mengetahui keadaan di luar lingkungan kita. Dengan adanya teknologi kita memudahkan informasi yang kita butuhkan, tapi untuk informasi tertentu mungkin kita tidak mendapatkannya. Misalnya, informasi mengenai keadaan di lingkungan kita, karena keadaan yang berbeda di setiap tempat keadaan lingkungan berbeda.

Oleh karena itu banyak yang menciptakan alat untuk mengetahui keadaan lingkungan sekitarnya, misalnya dengan pembuatan alat pengukur arah angin, kecepatan angin dan suhu yang biasanya digunakan untuk bandara.

PERMASALAHAN

Pengukuran arah angin yang ada sekarang ini hanya mengukur dalam satuan Utara, Selatan, Barat, Timur dan arah yang lainnya. Sehingga kita tidak tahu arah angin secara detail (dalam satuan derajat). Dengan menggunakan Motor Stepper kita dapat mengetahui arah angin dengan satuan derajat dengan bantuan sensor optocoupler.

dan buah sensor suhu LM35, yang berfungsi untuk mengukur suhu udara.

MAKSUD DAN TUJUAN

Maksud dan tujuan dari pembuatan alat ini adalah untuk membuat alat pengukur Arah Angin, Kecepatan Angin dan Suhu Udara yang dimonitor oleh komputer melalui PPI 8255.

TEMATIKA PEMBAHASAN

Sistematika pembahasan Tugas Akhir ini dilakukan sebagai berikut :

- : Membahas tentang PENDAHULUAN yang terdiri dari latar belakang, maksud dan tujuan, permasalahan, batasan masalah dan sistematika pembahasan.
- : Membahas tentang TEORI PENUNJANG yang berhubungan dengan komponen alat pengukur arah angin, kecepatan angin dan suhu udara.
- I : Membahas tentang PERENCAAN ALAT yang terdiri dari cara kerja alat yang dibuat dalam masing-masing bagiannya dan secara keseluruhan.
- V : Membahas tentang PENGUJIAN ALAT.
- V : Merupakan PENUTUP yang terdiri dari kesimpulan dan saran-saran yang diharapkan.
- V : Merupakan PENUTUP yang terdiri dari kesimpulan dan saran-saran yang diharapkan.

BAB II

TEORI PENUNJANG

PROGRAMMABLE PERIPHERAL INTERFACE 8255 (PPI 8255A)⁽³⁾

PPI 8255 merupakan peripheral interface yang dapat diprogram fungsinya. PPI ini menjadi dua group, group A yang terdiri dari Port A (PA0 - PA7) dan Port C Upper (PC7), dan group A yang terdiri dari Port B (PB0 - PB7) dan Port C Lowwer (PC0 - PC7). Untuk mengaktifkan mode 0, mula-mula pemrogram mengeset Control Word pada Register (CR). PPI mempunyai jumlah pin sebanyak 40 buah dan mempunyai jalur input/output (I/O) sebanyak 24 jalur. Dimana masing-masing pin telah mempunyai fungsi sendiri.

fungsi tersebut adalah :

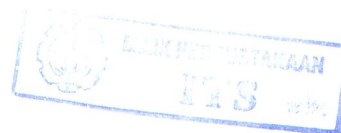
a Bus (D0 - D7)

digunakan untuk input atau output dari peralatan luar, dimana semua informasi diterima dan dikirimkan melalui 8 bit data.

Chip Select (CS)

ini digunakan untuk mengaktifkan chip 8255, bila mendapat logika "0" komputer akan mengirim data atau menerima data dari IC 8255.

Read (RD)



ombinasi dari kedua address input ini menentukan register mana dari PPI yang akan menerima atau mengirim data dari atau ke komputer.

et

gsi pin ini adalah untuk me-reset PPI 8255 dengan memberikan input logika "1". Pada

ini semua I/O port diatur dalam mode input.

t A (PA0 – PA7)

t ini digunakan sebagai 8 bit input/output untuk berhubungan dengan perangkat luar.

t B (PB0 – PB7)

erti port A, tetapi keduanya tidak saling berhubungan.

t C (PC0 – PC7)

na sepert Port A dan Port B, tetapi pada port C dibagi menjadi dua kelompok yaitu port

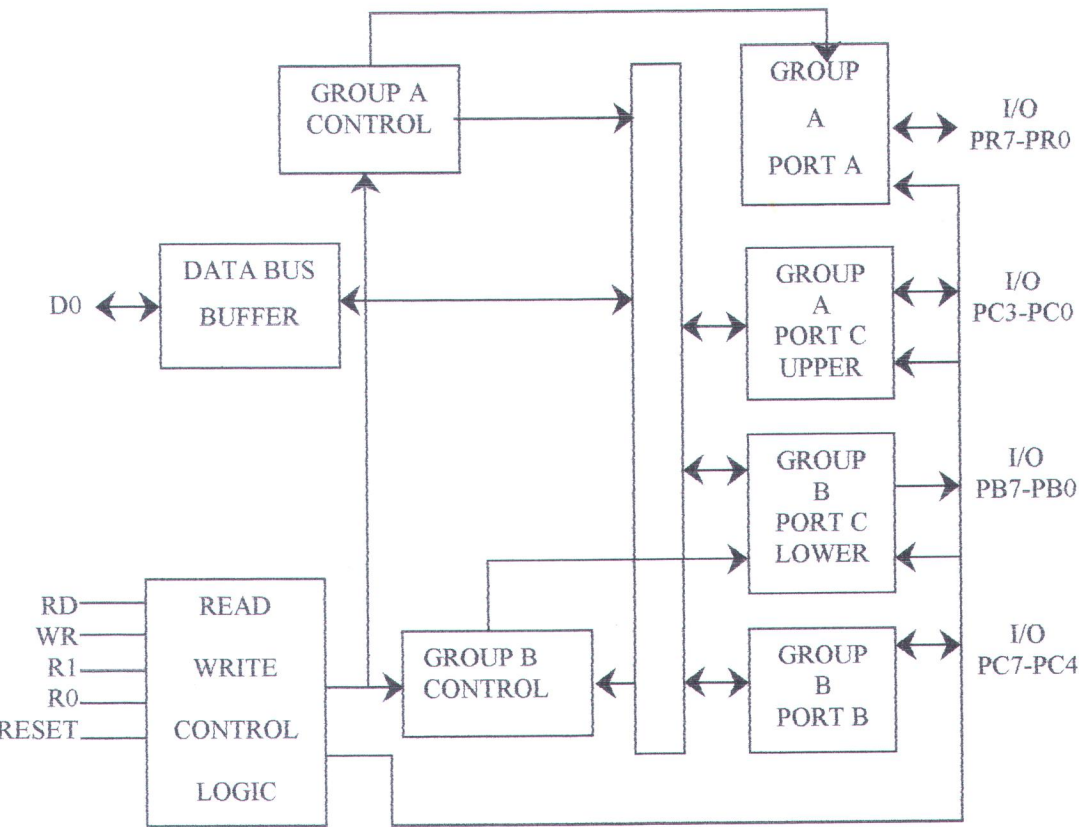
pper (PC0 – PC 3) dan lower (PC4 – PC 7).

c

agai sumber tegangan +5 Volt.

D (Ground)

tanahan dengan besar tegangan 0 Volt.



Gambar 2.1 Blok Diagram PPI 8255

PENGATURAN MODE PPI 8255

PI 8255 juga mempunyai 3 mode operasi, yaitu :

- de 0
- de operasi dimana semua port dapat dioperasikan sebagai masukan atau keluaran.
- tika port akan difungsikan sebagai masukan atau keluaran tanpa strobe, maka 8255

de 2

de operasi dengan menggunakan port A bekerja sebagai masukkan atau keluaran dua
n (bidirectional), juga untuk menerima masukan dan mengeluarkan data, maka port
isialisasikan pada mode 2.

Dibawah ini merupakan tabel yang menunjukkan dasar-dasar operasi PPI 8255, yaitu:

Tabel 2.1 Operasi Dasar PPI 8255

A1	A0	RD	WR	CS	operasi input (READ)	
0	0	0	1	0	Port A	→ Data Bus
0	1	0	1	0	Port B	→ Data Bus
1	0	0	1	0	Port C	→ Data Bus
					Operasi Output (WRITE)	
0	0	1	0	0	Data Bus	→ Port A
0	1	1	0	0	Data Bus	→ Port B
1	0	1	0	0	Data Bus	→ Port C
1	1	1	0	0	Data Bus	→ Control
					Fungsi Disable	
X	X	X	X	1	Data Bus	→ 3 - State
1	1	0	1	0	Kondisi Ilegal	
X	X	1	1	0	Data Bus	→ 3 - State

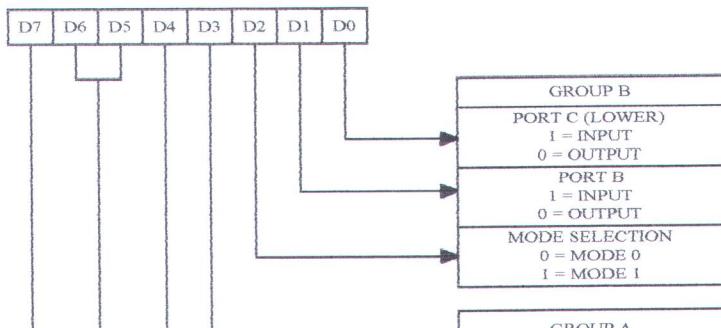
Mode 0 memungkinkan operasi input output sederhana pada masing-masing dari
port (port A, port B, dan port C).

si fungsional Mode 0:

edini dua port masing-masing 8 bit (port A dan port B)

Tabel 2.2 Mode Pengoperasian PPI 8255

A		B		GROUP A			GROUP B	
D4	D3	D1	D0	PORT A	PORT C (UPPER)	#	PORT B	PORT C (LOWER)
0	0	0	0	OUTPUT	OUTPUT	0	OUTPUT	OUTPUT
0	0	0	1	OUTPUT	OUTPUT	1	OUTPUT	INPUT
0	0	1	0	OUTPUT	OUTPUT	2	INPUT	OUTPUT
0	0	1	1	OUTPUT	OUTPUT	3	INPUT	INPUT
0	1	0	0	INPUT	INPUT	4	OUTPUT	OUTPUT
0	1	0	1	INPUT	INPUT	5	OUTPUT	INPUT
0	1	1	0	INPUT	INPUT	6	INPUT	OUTPUT
0	1	1	1	INPUT	INPUT	7	INPUT	INPUT
1	0	0	0	OUTPUT	OUTPUT	8	OUTPUT	OUTPUT
1	0	0	1	OUTPUT	OUTPUT	9	OUTPUT	INPUT
1	0	1	0	OUTPUT	OUTPUT	10	INPUT	OUTPUT
1	0	1	1	OUTPUT	OUTPUT	11	INPUT	INPUT
1	1	0	0	INPUT	INPUT	12	OUTPUT	OUTPUT
1	1	0	1	INPUT	INPUT	13	OUTPUT	INPUT
1	1	1	0	INPUT	INPUT	14	INPUT	OUTPUT
1	1	1	1	INPUT	INPUT	15	INPUT	INPUT



agan :

(Mode Set Flag)

fungsi untuk menentukan aktif tidaknya PPI 8255, bila aktif bernilai "1", tidak aktif bernilai "0".

dan D5 mode (Selectiion Group A)

fungsi untuk menentukan mode operasi kelompok A

de 0 = 00

de 1 = 01

de 2 = 1X (X=1, X = 0)

(port A)

fungsi untuk menentukan fungsi port A

ut = "1"

tput = "0"

(port C upper)

fungsi untuk menentukan operasi port C upper

out = "1"

input = "0"

(Mode Selection Grooup B)

fungsi untuk menentukan mode operasi kelompok B

(port C Lower)

fungsi untuk menentukan fungsi port C lower

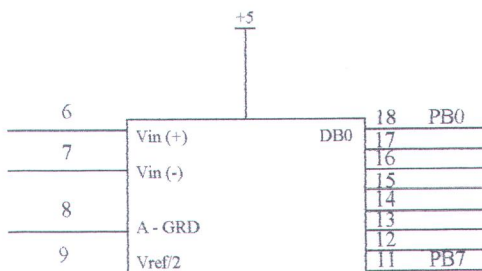
ut = "1"

put = "0"

Ketika PPI mendapat sinyal reset, maka port diset menjadi mode input. Setelah inisialisasi pada PPI 8255 tersebut dapat ditentukan apakah berfungsi sebagai input atau output.

ANALOG TO DIGITAL KONVERTER (ADC)⁽⁵⁾

ADC bertujuan untuk mengubah tegangan analog yang berasal dari input menjadi besaran digital 8 bit. ADC yang digunakan adalah ADC type 0804 yang mempunyai 8 bit, yang menggunakan metode successive approximation. ADC 0804 didesain secara khusus agar dapat digunakan pada sistem yang berbasis mikroprosesor. Karena itu ADC 0804 memiliki pin-pin yang diperlukan oleh mikroprosesor yaitu Chip Select (CS), Read (RD), Write (WR). konfigurasi pin-pin ADC 0804 dapat dilihat pada gambar dibawah ini.



pasang pada kaki 19 dan kapasitor pada kaki 4 kemudian dihubungkan ke ground. Hal ini akan mempengaruhi frekuensi yang ditentukan dengan rumus:

$$F = \frac{1}{(1,1 \times R \times C)}$$

$$R = 10K$$

$$C = 150pF$$

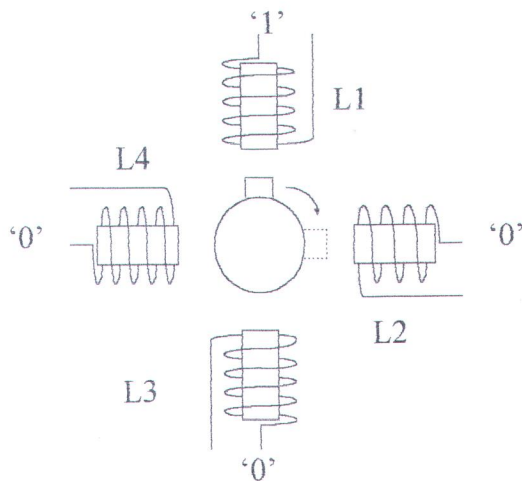


Dengan demikian frekuensi dari ADC 0804 sebesar 600 KHz. Sedangkan frekuensi ini akan dihubungkan ke kaki 4 (CLK-IN). Tegangan input analog masuk melalui kaki Vin(+) dan Vin(-) dapat dipakai untuk mengurangi besarnya tegangan pada input analog tersebut agar tidak melebihi tegangan Vcc. Kaki 9 (Vref/2) dihubungkan ke tegangan yang besarnya sama dengan referensi, tegangan input Vref/2 ini perlu untuk akurasi dari hasil output yang akan dihasilkan.

MOTOR STEPPER⁽⁴⁾

Motor stepper adalah jenis motor yang dapat digerakkan dengan step sudut yang tertentu. Ketika arus putaran stator diubah. Pengaturan posisi putaran sesuai dengan pulsa yang diberikan. Kecepatan dan pengaturan kecepatan putar diatur oleh perubahan kecepatan pulsa persatuan waktu.

Logika perputaran rotor pada motor stepper dapat dianalogikan secara langsung dengan data "0" atau "1" yang diberikan secara serentak terhadap masing-masing lilitan pada



Gambar 2.4 Prinsip Kerja Motor Stepper

Pada prinsipnya ada dua macam kerja motor stepper, yaitu Full Step dan Half Step.
bel dibawah ini.

Tabel 2.3 (a) Logika untuk putaran Full Step

Full Step								
STEP	Berputar Searah Jarum Jam				Berputar Berlawanan arah Jarum Jam			
	L1	L2	L3	L4	L1	L2	L3	L4
1	1	0	0	0	0	0	0	1
2	0	1	0	0	0	0	1	0
3	0	0	1	0	0	1	0	0
4	0	0	0	1	1	0	0	0

Tabel 2.3 (b) Logika untuk putaran Half Step

Half Step								
STEP	Berputar Searah Jarum Jam				Berputar Berlawanan arah Jarum Jam			
	L1	L2	L3	L4	L1	L2	L3	L4
1	1	0	0	0	0	0	0	1
2	1	1	0	0	0	0	1	1
3	0	1	0	0	0	0	1	0
4	0	1	1	0	0	1	1	0
5	0	0	1	0	0	1	0	0
6	0	0	1	1	1	1	0	0
7	0	0	0	1	1	0	0	0
8	1	0	0	1	1	0	0	1

Untuk memperoleh efek cengkaman yang lebih kuat, modus data yang diberikan pada Full wave dapat dimanupulasi dengan memberikan double active bit pada setiap formasi (tabel 2.4). dengan cara ini torsi yang dihasilkan akan lebih besar.

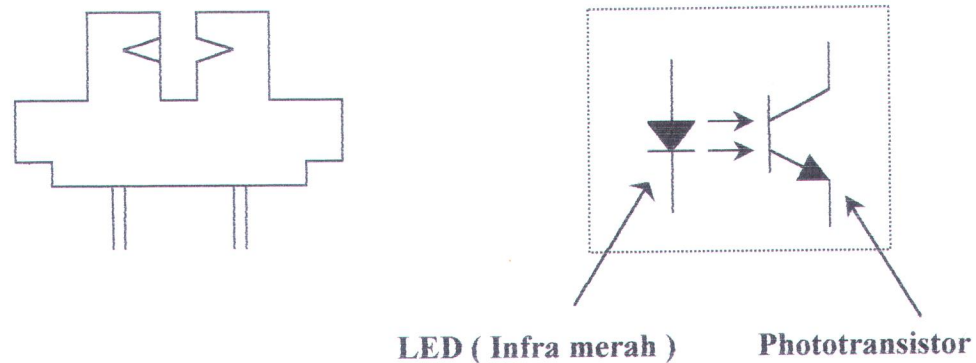
Pada Full Step, suatu titik pada sebuah kutub magnet di rotor akan kembali mendapatkan medan magnet stator pada lilitan yang sama setelah step ke-4. Kemudian dapat di mulai lagi mulai dari step pertama. Setiap step, rotor bergerak baik searah jarum jam atau berlawanan jarum jam sebesar $\frac{1}{4}$ dari 360° dibagi dengan jumlah kutub magnet (batangan besi) yang berada disisi rotor.

Untuk Half Step, setiap kutub magnet pada rotor akan kembali mendapat tarikan dari

magnet di rotor diperbanyak dan lilitan dibuat berpasang-pasangan sesuai dengan kutub magnet rotor.

OPTOCOUPLER⁽⁴⁾

Optocoupler merupakan gabungan kata dari *Opto/Optic* dan *coupler* yaitu komponen yang menghubungkan (*coupling*) yang berdasarkan picu cahaya.. Optocoupler terdiri dari dua bagian yaitu transmitter dan receiver, transmitter terdiri dari Led Infra Merah, receiver terdiri dari transistor yang akan memperoleh bias maju bila mendapat sinar dari led transmitter.

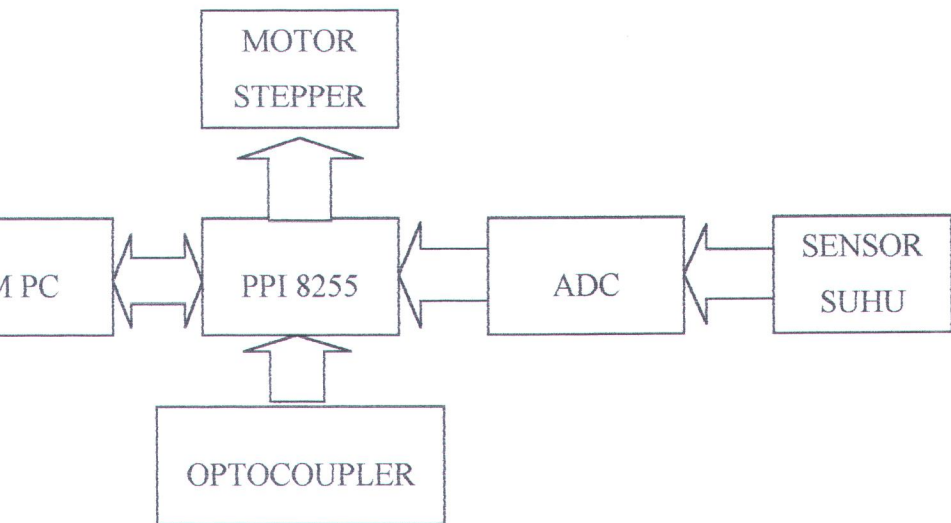


Gambar 2.5 Fisik OptoCoupler dan isi rangkaianannya

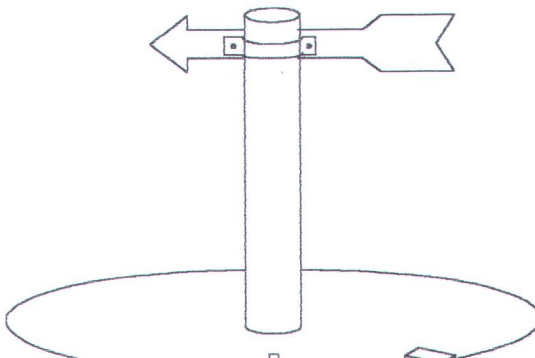
BAB III

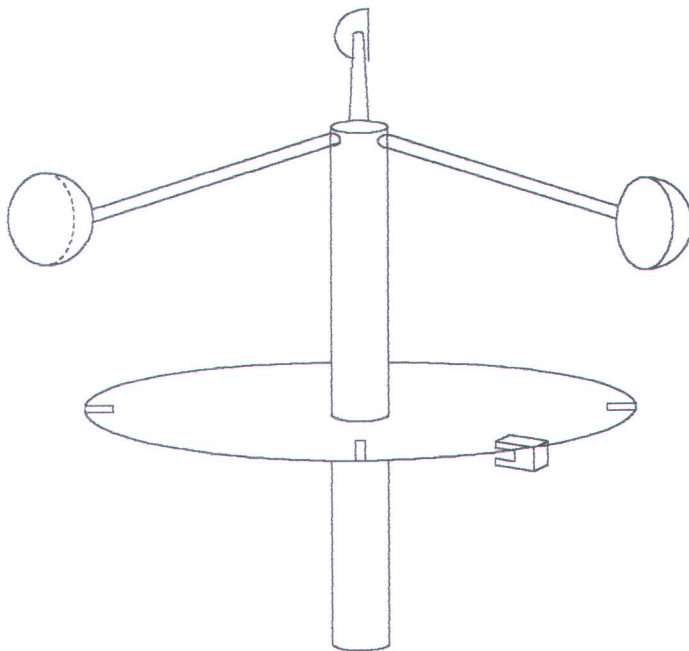
PERENCANAAN ALAT

BLOK DIAGRAM ALAT



Gambar 3.1 Blok Diagram Alat





Gambar 3.3 Bentuk Fisik Alat Pengukur Kecepatan Angin

PERENCANAAN PERANGKAT KERAS

1. RANGKAIAN ADC

Rangkaian ADC ini bertujuan untuk mengubah tegangan analog menjadi tegangan

. Untuk melakukan konversi, terlebih dahulu CS diberikan logika “0” sehingga data

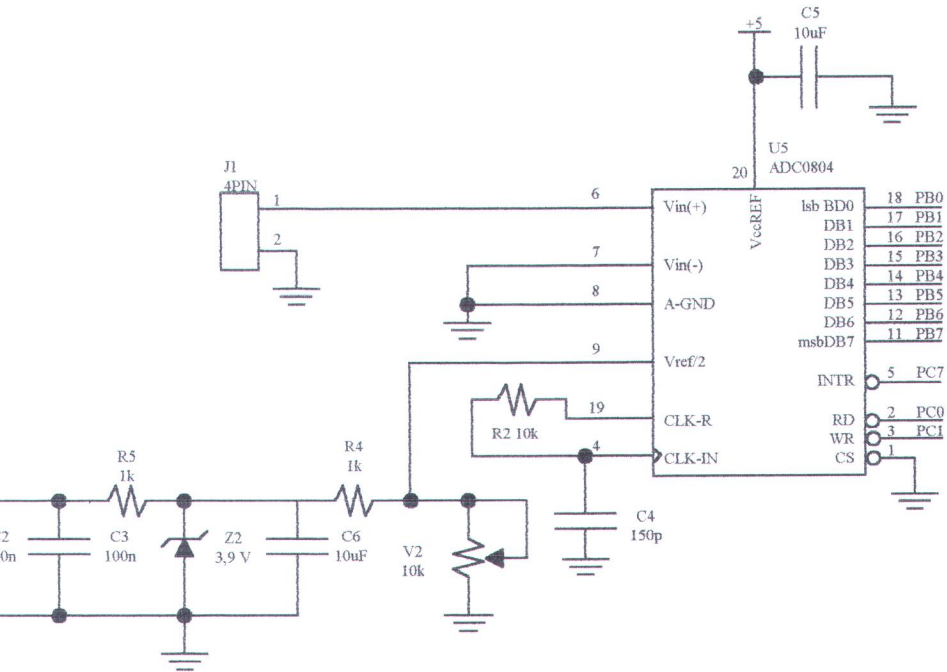
dibaca oleh ADC 0804. Kemudian WR diberikan logika “0”, sehingga ADC 0804

lai melakukan konversi. Setelah konversi selesai WR diberikan logika “1” dan RD

kan logika “0” maka output dapat terbaca.

Clock dari ADC menggunakan pembangkit internal sehingga hanya dibutuhkan sebuah

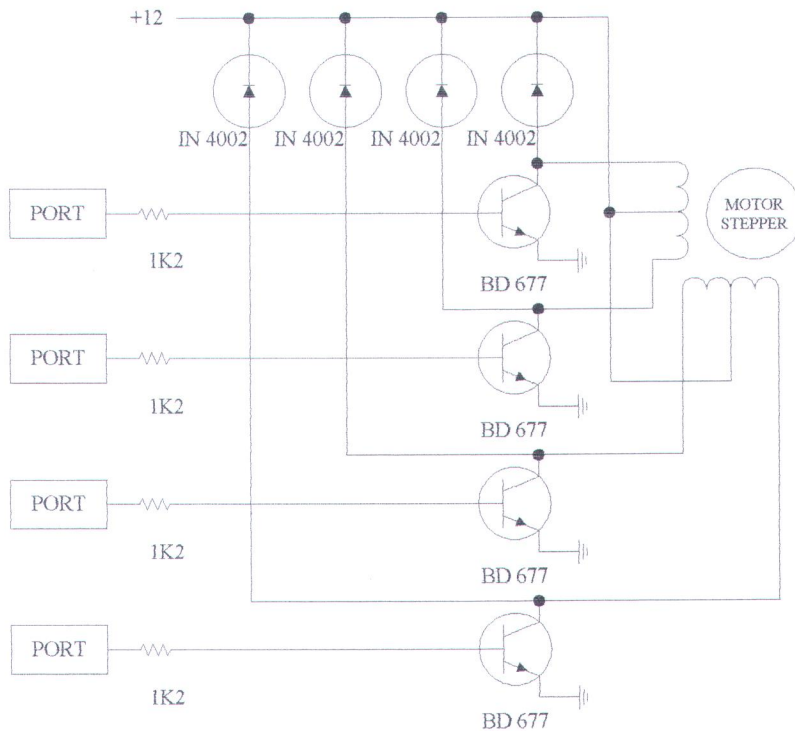
nsi masih berada dalam batas range ADC yaitu ADC 0804 antara 100 KHz sampai KHz.



Gambar 3.4 Rangkaian ADC

RANGKAIAN DRIVER MOTOR STEPPER

Rangkaian ini membutuhkan tegangan supply sebesar 12 Volt untuk dapat menggerakkan motor stepper. Komponen-komponen yang digunakan adalah transistor BD 677, resistor 1,2 Kilo Ohm, dan dioda IN 4002. Pin yang dibutuhkan untuk mengontrol putaran motor adalah 4 pin dari port A. Gambar rangkaian driver motor stepper dapat dilihat pada Gambar 3.5.



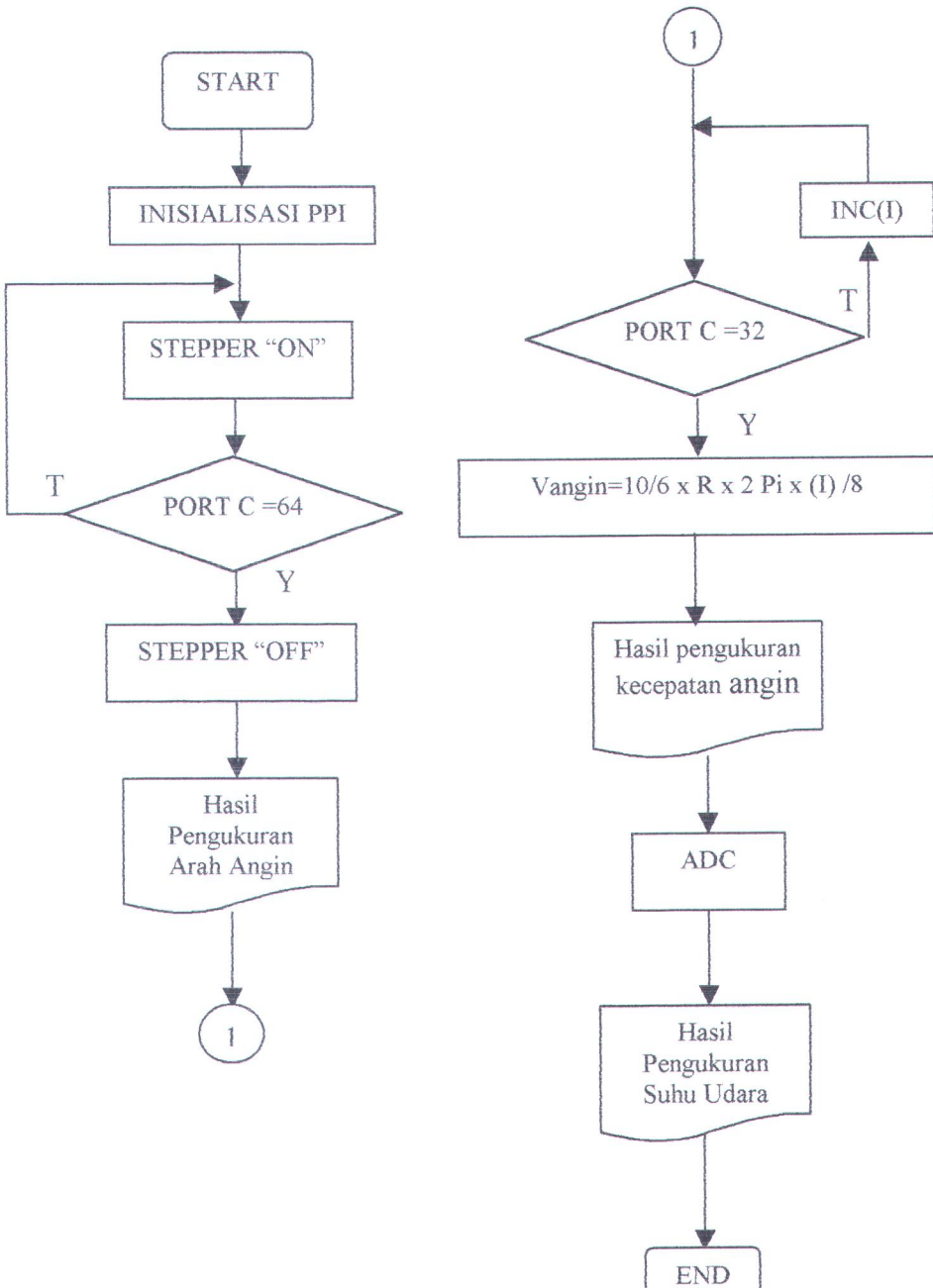
Gambar 3.5 Rangkaian Driver Motor Stepper

RANGKAIAN OPTOCOUPLER

Optocoupler terdiri dari dua bagian yaitu pemancar dan penerima (sensor), setiap membutuhkan tegangan supply sebesar +5 dan diberikan tahanan sebesar 330 Ohm. Sinyal dari sensor diberikan ke basis transistor BC 548 yang kaki emiter diberikan tahanan sebesar 4,7 Kilo Ohm, tahanan inilah yang tegangannya diukur.



FLOWCHART



BAB IV

PENGUJIAN ALAT

Pengujian alat dalam penyelesaian sangat penting, karena dengan pengujian alat ini dapat mengetahui apakah alat yang kita buat bekerja dengan baik. Pengujian tersebut meliputi pengujian PPI 8255, rangkaian ADC, rangkaian optocoupler, rangkaian suhu, dan rangkaian relay.

PENGUJIAN PPI 8255

Pengujian alat ini dilakukan dengan menjalankan program sederhana, dengan cara tersebut adalah :

memberikan kontrol word register untuk inisialisasi PPI 8255, nilai control word yang menginisialisasi semua port sebagai output.

Mengaktifkan semua port dengan memberikan nilai pada tiap kaki yang terhubung dengan port A, port B, dan port C.

Program tersebut adalah:

```
Uses Crt;  
Begin
```

```
    Port[$30F]:=$80;
```

```
    Port[$30C]:=255;
```

```
    Port[$30D]:=255;
```

```
    Port[$30E]:=255;
```

Tabel 4.1 Hasil Keluaran Logika “1” PPI 8255

PIN	PORT A (Volt)	PORT B (Volt)	PORT C (Volt)
PIN 1	4,2	4,2	4,2
PIN 2	4,2	4,2	4,2
PIN 3	4,2	4,2	4,2
PIN 4	4,2	4,2	4,2
PIN5	4,2	4,2	4,2
PIN 6	4,2	4,2	4,2
PIN 7	4,2	4,2	4,2
PIN8	4,2	4,2	4,2

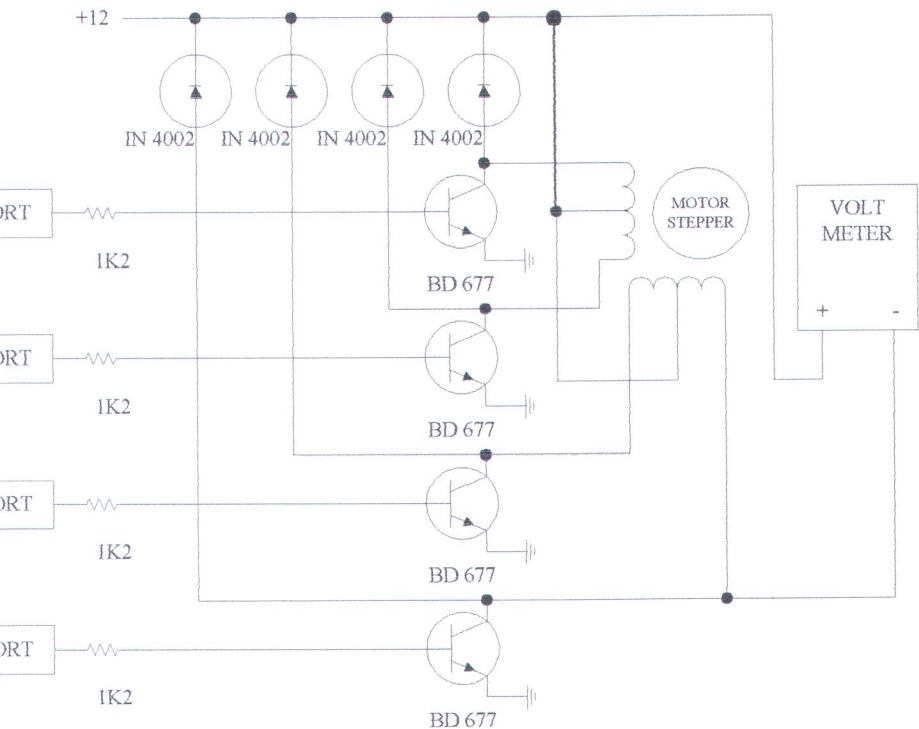
Tabel 4.2 Hasil Keluaran Logika “0” PPI 8255

PIN	PORT A (Volt)	PORT B (Volt)	PORT C (Volt)
PIN 1	0,01	0,01	0,01
PIN 2	0,01	0,01	0,01
PIN 3	0,01	0,01	0,01
PIN 4	0,01	0,01	0,01
PIN5	0,01	0,01	0,01
PIN 6	0,01	0,01	0,01
PIN 7	0,01	0,01	0,01

tmeter pada tegangan power supply. Pada gambar 4.1 menunjukkan bahwa port A
pengaktif transistor, apabila salah satu kaki basis pada transistor diberikan logika “1”
Vcc semula pada kondisi close loop menjadi open loop. Pemberian logika diberikan
berurutan untuk dapat bergerak memutar, gerakan motor stepper bergerak sedikit demi

Pengujian diatas kita menggunakan program yaitu:

```
t;  
Data : array [1..4] of byte;  
step : integer;  
  
Clrscr;  
Step:=1;  
Port[$30f]:=$80;  
Data[1]:=$1; Data[2]:=$2; Data[3]:=$4; Data[4]:=$8;  
Repeat  
    Port[$30c]:=data[step];  
    Delay(2);  
    Inc(step);  
    If step=5 then step:=1;  
Until keypressed;  
Step:=4;  
Repeat  
    Port[$30c]:=data[step];  
    Delay(2);  
    Dec(step);  
    If step=0 then step:=4;  
Until keypressed;  
End.
```



Gambar 4.1 Pengujian Rangkaian Driver Motor Stepper

ENGUJIAN RANGKAIAN OPTOCOUPLER

Pengujian rangkaian ini dilakukan dengan menghalangi celah yang ada pada optocoupler, yaitu apabila terhalang maka logic bernilai ‘0’, sebaliknya apabila tidak terhalang maka logic bernilai ‘1’.

Tabel 4.3 Hasil Pengukuran Rangkaian Optocoupler

Output 1	Kondisi	Volt
----------	---------	------

Tabel 4.4 Data Pengukuran LM35

SUHU (° C)	Pengukuran (m Volt)		
	I	II	III
30	295	301	308
40	385	400	410
50	496	502	512
60	588	605	616
70	680	710	709
80	795	802	811
90	895	900	907
100	990	1005	1009

BAB V

PENUTUP

Setelah melakukan uji coba terhadap alat baik perangkat keras (Hardware) maupun perangkat lunak (Software), maka diperoleh kesimpulan dan saran yang nantinya diharapkan untuk menambah ilmu dan teknologi.

KESIMPULAN

Dari hasil pengujian dapat diambil kesimpulan sebagai berikut :

Pengukuran Arah Angin

Pada alat ini telah bekerja cukup baik, dimana alat ini telah dapat mengetahui keadaan arah angin.

Putaran motor stepper begitu lambat, namun untuk perubahan arah angin cukup lama sehingga alat ini cukup memadai untuk pengukuran arah angin.

Keadaan angin yang sangat kecil sehingga tidak cukup untuk mendorong lempengan.

Pengukuran Kecepatan Angin

Keadaan angin yang kecil sehingga angin tidak mampu untuk memutar kincir, ini mungkin dapat diperbaiki dengan memperbesar cangkang, yang harus sesuai dengan jari-jari kincir.

SARAN

Dengan adanya kekurangan-kekurangan pada alat yang direncanakan, saran dan untuk pengembangan alat yang mungkin dilakukan adalah :

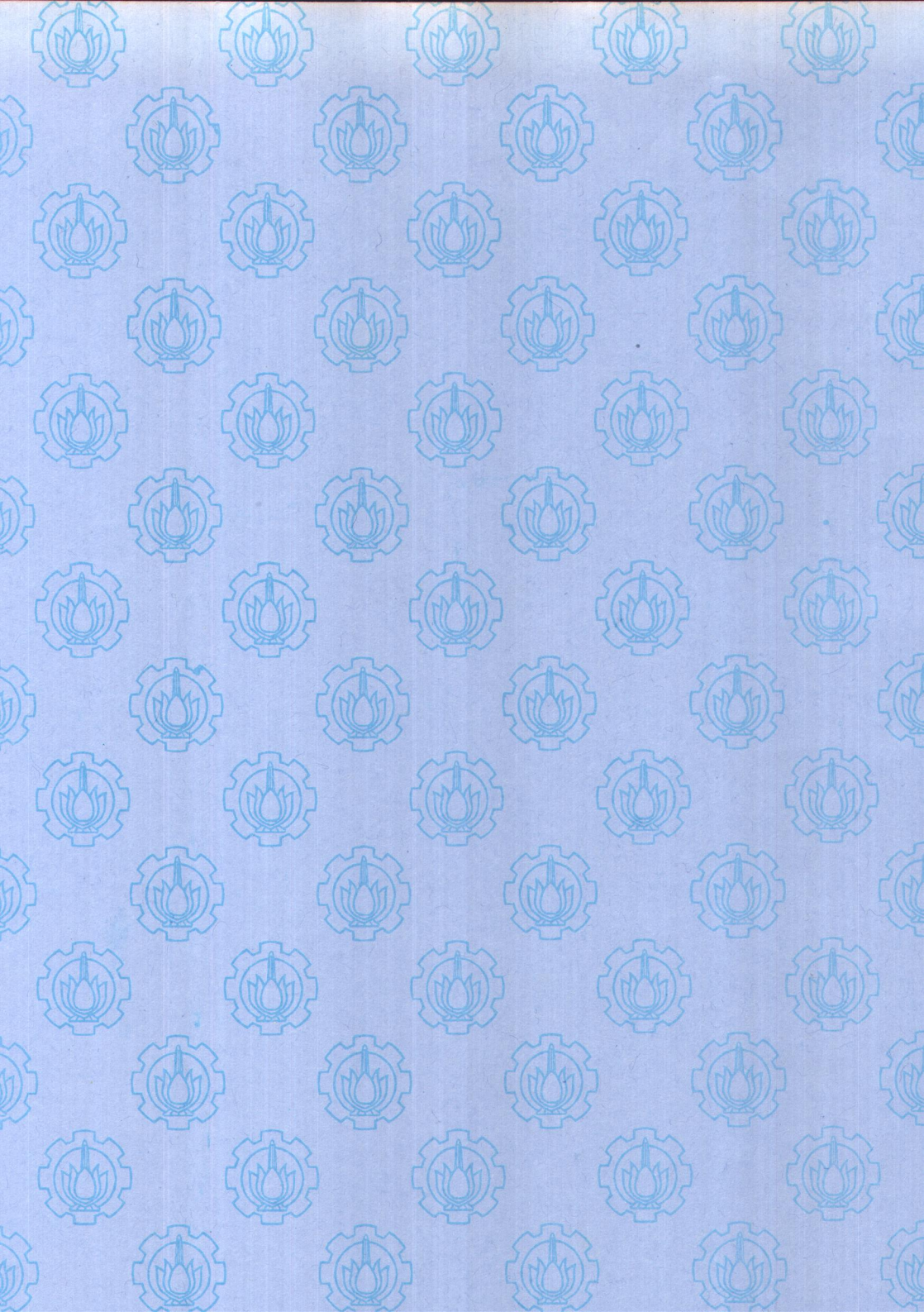
Dalam sistem pengukuran arah angin, kecepatan angin dan suhu dapat menggunakan media pengontrol dari mikrokontroler, sehingga dapat mengefisienkan biaya dan tempat dengan dibantu mikrokontroller.

Pada pengukuran arah angin disarankan menggunakan sensor yang lain, yang lebih presisi dan lebih cepat.

DAFTAR PUSTAKA

- ntifikasi Tenaga Angin Hicromicro Jawa Timur – Bali, kerjasama PUSLIK ITS & MDA JATIM - BALI, 1983.
- ianto H.M, Turbo Pascal Jilid I, Andi offset, Yogyakarta, 1993
- M. Steeman, Data Sheet Book 2, Penerbit PT. Elek Media Komputindo, Kelompok media, Jakarta 1992.
- rkoprosesor & Interface I, Politeknik Elektronika Surabaya, 1990.
- tional Semiconductor, Linear Data Sheet Book, Santa Clara, California.





PIRAN

LISTING PROGRAM

0000;

s, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
Dialogs, StdCtrls, ExtCtrls, Mask;

```
1 = class(TForm)
```

```
1: TTimer;
```

```
2: TTimer;
```

```
h1: TButton;
```

```
h2: TButton;
```

```
3: TTimer;
```

```
h3: TButton;
```

```
4: TTimer;
```

```
4: TLabel;
```

```
5: TTimer;
```

```
6: TTimer;
```

```
1: TBevel;
```

```
2: TBevel;
```

```
5: TLabel;
```

```
5: TLabel;
```

```
7: TLabel;
```

```
8: TLabel;
```

```
9: TLabel;
```

```
10: TLabel;
```

```
11: TLabel;
```

```
12: TLabel;
```

```
13: TLabel;
```

```
14: TLabel;
```

```
7: TTimer;
```

```
16: TLabel;
```

```
1: TLabel;
```

```
10: TTimer;
```

```
11: TTimer;
```

```
procedure Timer1Timer(Sender: TObject);
```

```
procedure Timer2Timer(Sender: TObject);
```

```
procedure Button1Click(Sender: TObject);
```

```
procedure Button2Click(Sender: TObject);
```

```
procedure Timer3Timer(Sender: TObject);
```

```
procedure Timer4Timer(Sender: TObject);
```


array [1..10] of integer;
integer;
ing;
boolean;

data,derajat:real;

ntation

FM}

re TForm1.Timer1Timer(Sender: TObject);

.enabled:=false;

hen i:=1;
00]:=data1[i];

[\$302];
d 64;
ai=false) or (x=64) or (j=436) then

lai=false) then

on2.enabled:=false;
r1.enabled:=false;
r2.enabled:=false;
r3.enabled:=true;

36) then

on2.enabled:=false;
r1.enabled:=false;
r2.enabled:=false;
r3.enabled:=true;

4 then

jat:=j*36/43.6;
erajat:5:2,kata);
l.caption:=kata+ ' °';
bargrafik(sender);
on2.enabled:=false;
r1.enabled:=false;
r2.enabled:=false;

```
[$302];  
d 64;  
ai=false) or (x=64) or (j=436) then
```

```
ai=false then
```

```
on2.enabled:=false;  
r1.enabled:=false;  
r2.enabled:=false;  
r4.enabled:=true;
```

```
36) then
```

```
on2.enabled:=false;  
r1.enabled:=false;  
r2.enabled:=false;  
r4.enabled:=true;
```

```
4 then
```

```
at:=j*36/43.6;  
erajat:5:2,kata);  
r1.caption:=kata+'°';  
pagrafik(sender);  
on2.enabled:=false;  
r1.enabled:=false;  
r2.enabled:=false;  
r4.enabled:=true;
```

```
r2.enabled:=false;  
r1.enabled:=true;
```

```
re TForm1.Button1Click(Sender: TObject);
```

```
03]:=$8a;  
=false;  
enabled:=false;  
enabled:=false;
```

```
re TForm1.Timer3Timer(Sender: TObject);
```

```
    when i:=8;  
    00]:=data1[i];
```

```
    when
```

```
    :=false;  
    3.enabled:=false;  
    4.enabled:=false;  
    1.enabled:=true;  
    2.enabled:=true;  
    0.enabled:=true;
```

```
    3.enabled:=false;  
    4.enabled:=true;
```

```
re TForm1.Timer4Timer(Sender: TObject);
```

```
    when i:=8;  
    00]:=data1[i];
```

```
    when
```

```
    :=false;  
    3.enabled:=false;  
    4.enabled:=false;  
    1.enabled:=true;  
    2.enabled:=true;  
    0.enabled:=true;
```

```
    4.enabled:=false;  
    3.enabled:=true;
```

```

]:=4;data1[6]:=12;data1[7]:=8;data1[8]:=72;
.Pen.Color := clblack;
.Pen.Width := 2;
.Pen.Mode := pmcopy;
.Brush.Color := clblack;
.Brush.Style := bssolid;
Rectangle(400,300,700,501);
Rectangle(400,50,700,251);
0;k:=0;l:=0;
enabled:=true;

```

```

re TForm1.gambargrafik(Sender: TObject);

```

```

.Pen.Color := clblack;
.Pen.Width := 2;
.Pen.Mode := pmcopy;
.Brush.Color := clsilver;
.Brush.Style := bssolid;
.Ellipse (50,120,350,420);
.Brush.Color := clblack;
.Pen.Color := clblack;
.Pen.Width := 2;
.Pen.Mode := pmcopy;
.Brush.Color := clgray;
.Brush.Style := bssolid;
.Ellipse (190,260,210,280);
moveto(50,270);
lineto(350,270);
moveto(200,120);
lineto(200,420);
.Pen.Color :=clblue;
.Pen.Width := 2;
:= pi * derajat/180;
s.moveTo (200,270);
s.lineTo (
und(140*(sin(buffer))),
und(-140*(cos(buffer))));

```

```

re TForm1.FormCreate(Sender: TObject);

```

```

003]:=$8a;
2.enabled:=false;

```



```

Pen.Color := clblack;
Pen.Width := 2;
Pen.Mode := pmcopy;
Brush.Color := clblack;
Brush.Style := bssolid;

```



```

302]:=15;
302]:=0;
302]:=15;

```

```

=port[$302];
cek and 128;
cek<>128;
=port[$301];
=(data2*20)/1000;
100;
0;
data2*100+2;
1:2:0,kata);
caption:=kata + ' °C';
then

```

```

s.Rectangle(400,300,700,500);

```

```

nat[I]:=trunc(hasil);
Pen.Color := clblue;
Pen.Width := 1;
moveto(400+((I-1)*30),500-koordinat[I]);
then canvas.moveto(400+((I-1)*30),500-koordinat[I]);
lineto(400+(I*30),500-koordinat[I]);
0.enabled:=false;
1.enabled:=true;

```

```

re TForm1.Timer1.Timer(Sender: TObject);
menit,detik,milidetik:word;
1,waktu2:word;
m:integer;
kata2,kata3:string;

```

```

=true;

```

```

:=detik+1;
waktu1>=60 then waktu1:=waktu1-60;

Time (Now, Jam, Menit, Detik, MiliDetik);
:=detik;
[$302];
d 32;
then

lsa);

waktu2=waktu1;
then

s.Rectangle(400,50,700,251);

]:=trunc((10/6)*0.12*2*pi*(pulsa/8)/50);
.Pen.Color :=clred;
.Pen.Width := 1;
moveto(400+((k-1)*30),250-data2[k]);
hen canvas.moveto(400+((k-1)*30),250-data2[k]);
lineto(400+(k*30),250-data2[k]);
2[k],kata);
.caption:=kata + ' m/dt';
.enabled:=false;
.enabled:=true;

```

LISTING PROGRAM UTAMA

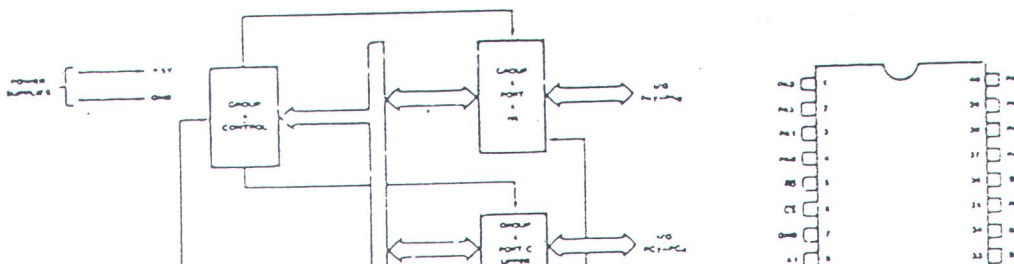
```
TA99;  
  
in 'TA2000.PAS' {Form1};  
  
ES}  
  
ation.CreateForm(TForm1, Form1);  
ation.Run;
```

LAMPIRAN

8255A/8255A-5 PROGRAMMABLE PERIPHERAL INTERFACE

- MCS-85™ Compatible 8255A-5
- 24 Programmable I/O Pins
- Completely TTL Compatible
- Fully Compatible with Intel® Microprocessor Families
- Improved Timing Characteristics
- Direct Bit Set/Reset Capability Easing Control Application Interface
- 40-Pin Dual In-Line Package
- Reduces System Package Count
- Improved DC Driving Capability

The Intel® 8255A is a general purpose programmable I/O device designed for use with Intel® microprocessors. It has 24 I/O pins which may be individually programmed in 2 groups of 12 and used in 3 major modes of operation. In the first mode (MODE 0), each group of 12 I/O pins may be programmed in sets of 4 to be input or output. In MODE 1, the second mode, each group may be programmed to have 8 lines of input or output. Of the remaining 4 pins, 3 are used for handshaking and interrupt control signals. The third mode of operation (MODE 2) is a bidirectional bus mode which uses 16 lines for a bidirectional bus, and 5 lines, borrowing one from the other group, for handshaking.



LAMPIRAN

8255A/8255A-5

PROGRAMMABLE PERIPHERAL INTERFACE

(RESET)

Reset. A "high on this input clears the control register and all ports (A, C, C) are set to the input mode.

Group A and Group B Controls

The functional configuration of each port is programmed by the systems software. In essence, the CPU "outputs" a control word to the 8255A. The control word contains information such as "mode", "bit set", "bit reset", etc., that initializes the functional configuration of the 8255A.

Each of the Control blocks (Group A and Group B) accepts "commands" from the Read/Write Control Logic, receives "control words" from the internal data bus and issues the proper commands to its associated ports.

Control Group A - Port A and Port C upper (C7-C4)

Control Group B - Port B and Port C lower (C3-C0)

The Control Word Register can Only be written into. No Read operation of the Control Word Register is allowed.

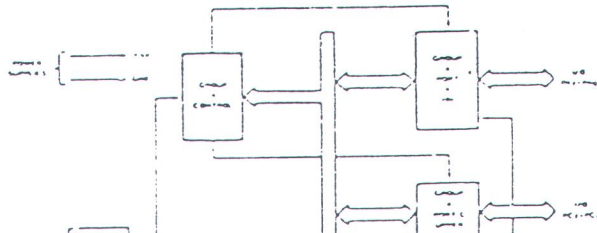
Ports A, B, and C

The 8255A contains three 8-bit ports (A, B, and C) which can be configured in a wide variety of functional characteristics by the system software but each has its own special features or "personality" to further enhance the power and flexibility of the 8255A.

Port A. One 8-bit data output latch/buffer and one 8-bit data input latch.

Port B. One 8-bit data input/output latch/buffer and one 8-bit data input buffer.

Port C. One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input). This port is divided into two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and it can be used for the control signal outputs and status signal inputs in conjunction with ports A and B.



PIN CONFIGURATION



LAMPIRAN

8255A/8255A-5
PROGRAMMABLE PERIPHERAL INTERFACE

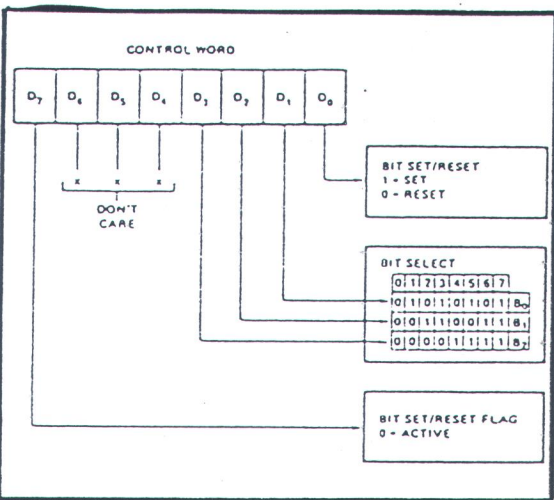


Figure 7. Bit Set/Reset Format

When Port C is being used as status/control for Port A or Port B, these bits can be set or reset by using the Bit Set/Reset operation just as if they were data output ports.

Interrupt Control Functions

When the 8255A is programmed to operate in mode 0 or mode 2, control signals are provided that can be used as interrupt request inputs to the CPU. The interrupt request signals, generated from port C, can be inhibited or enabled by setting or resetting the associated INTE flip-flop, using the bit set/reset function of port C.

This function allows the Programmer to disallow or allow a specific I/O device to interrupt the CPU without affecting any other device in the interrupt structure.

INTE flip-flop definition:

(BIT-SET) — INTE is SET — Interrupt enable

(BIT-RESET) — INTE is RESET — Interrupt disable

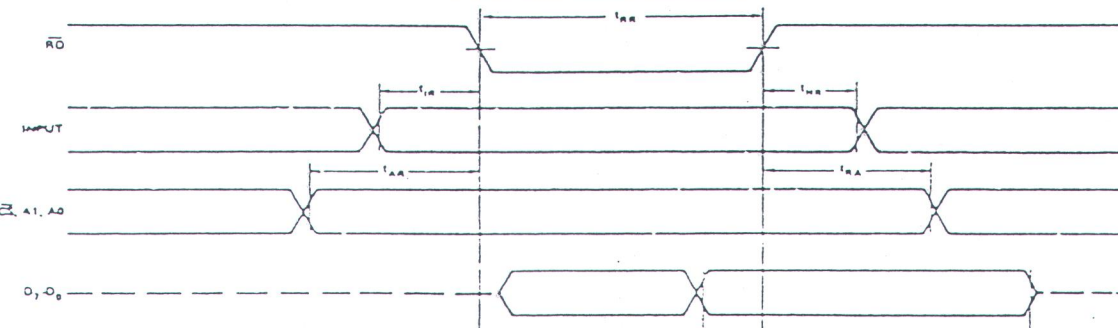
Note: All Mask flip-flops are automatically reset during mode selection and device Reset.

Operating Modes

MODE 0 (Basic Input/Output). This functional configuration provides simple input and output operations for each of the three ports. No "handshaking" is required, data is simply written to or read from a specified port.

Mode 0 Basic Functional Definitions.

- Two 8-bit ports and two 4-bit ports.
- Any port can be input or output.
- Outputs are latched.
- Inputs are not latched.
- 16 different Input/Output configurations are possible in this Mode.



LAMPIRAN

8255A/8255A-5

PROGRAMMABLE PERIPHERAL INTERFACE

Input Control Signal Definition

STB (Strobe Input). A "low" on this input loads data into the input latch.

IBF (Input Buffer Full FIF)

A "high" on this output indicates that the data has been loaded into the input latch; in essence, an acknowledgement. IBF is set by STB input being low and is reset by the rising edge of the RD input.

INTR (Interrupt Request)

A "high" on this output can be used to interrupt the CPU when an input device is requesting service. INTR is set by the STB is a "one", IBF is a "one" and INTE is a "one". It is reset by the falling edge of RD. This procedure allows an input device to request service from the CPU by simply strobing its data into the port.

INTE A

Controlled by bit set/reset of PC₄.

INTE B

Controlled by bit set/reset of PC₂.

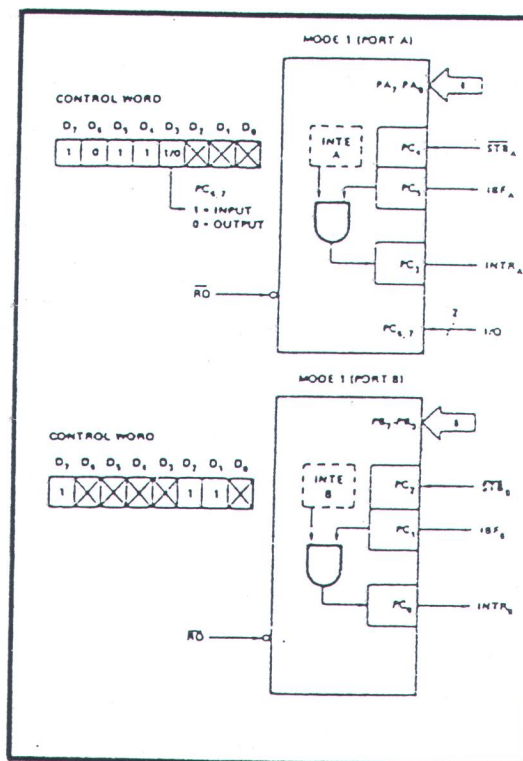


Figure 8. MODE 1 Input



8255A/8255A-5 PROGRAMMABLE PERIPHERAL INTERFACE

Output Control Signal Definition

OBF (Output Buffer Full FIF). The OBF output will go "low" to indicate that the CPU has written data out to the specified port. The OBF FIF will be set by the rising edge of the WR input and reset by ACK Input being low.

ACK (Acknowledge Input). A "low" on this input informs the 8255A that the data from port A or port B has been accepted. In essence, a response from the peripheral device indicating that it has received the data output by the CPU.

INTR (Interrupt Request). A "high" on this output can be used to interrupt the CPU when an output device has accepted data transmitted by the CPU. INTR is set when ACK is a "one", OBF is a "one" and INTE is a "one". It is reset by the falling edge of WR.

INTE A.

Controlled by bit set/reset of PC₆.

INTE B

Controlled by bit set/reset of PC₂.

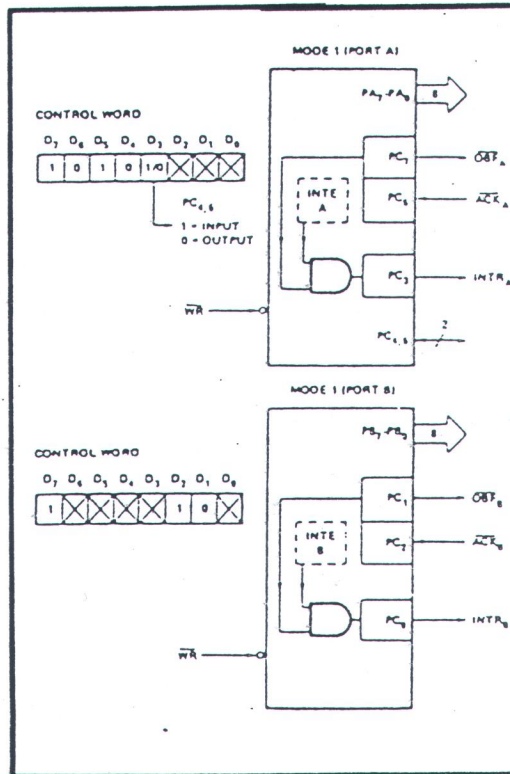


Figure 10. MODE 1 Output



LAMPIRAN

8255A/8255A-5

PROGRAMMABLE PERIPHERAL INTERFACE --

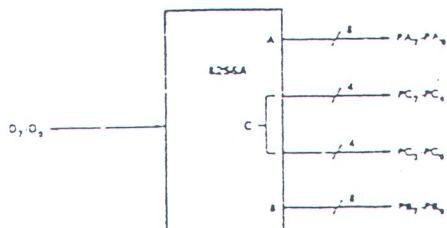
MODE 0 Port Definition

A		B		GROUP A			GROUP B	
D ₄	D ₃	D ₁	D ₀	PORT A	PORT C (UPPER)	#	PORT B	PORT C (LOWER)
0	0	0	0	OUTPUT	OUTPUT	0	OUTPUT	OUTPUT
0	0	0	1	OUTPUT	OUTPUT	1	OUTPUT	INPUT
0	0	1	0	OUTPUT	OUTPUT	2	INPUT	OUTPUT
0	0	1	1	OUTPUT	OUTPUT	3	INPUT	INPUT
0	1	0	0	OUTPUT	INPUT	4	OUTPUT	OUTPUT
0	1	0	1	OUTPUT	INPUT	5	OUTPUT	INPUT
0	1	1	0	OUTPUT	INPUT	6	INPUT	OUTPUT
0	1	1	1	OUTPUT	INPUT	7	INPUT	INPUT
1	0	0	0	INPUT	OUTPUT	8	OUTPUT	OUTPUT
1	0	0	1	INPUT	OUTPUT	9	OUTPUT	INPUT
1	0	1	0	INPUT	OUTPUT	10	INPUT	OUTPUT
1	0	1	1	INPUT	OUTPUT	11	INPUT	INPUT
1	1	0	0	INPUT	INPUT	12	OUTPUT	OUTPUT
1	1	0	1	INPUT	INPUT	13	OUTPUT	INPUT
1	1	1	0	INPUT	INPUT	14	INPUT	OUTPUT
1	1	1	1	INPUT	INPUT	15	INPUT	INPUT

MODE 0 Configurations

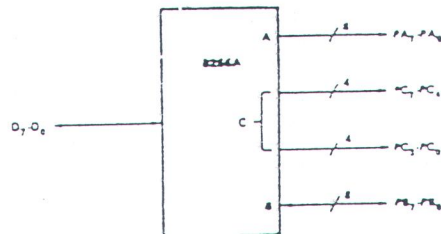
CONTROL WORD #0

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	0	0	0	0	0



CONTROL WORD #2

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	0	0	0	1	0

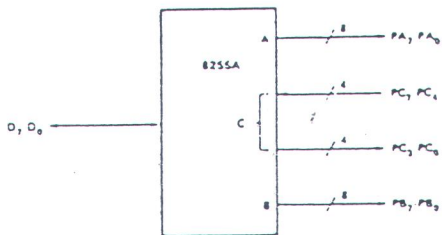


LAMPIRAN

8255A/8255A-5 PROGRAMMABLE PERIPHERAL INTERFACE

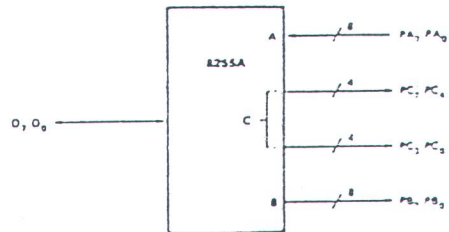
CONTROL WORD #4

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	0	1	0	0	0



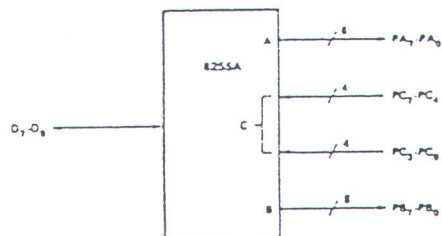
CONTROL WORD #8

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	1	0	0	0	0



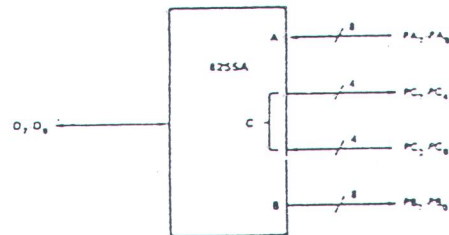
CONTROL WORD #5

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	0	1	0	0	1



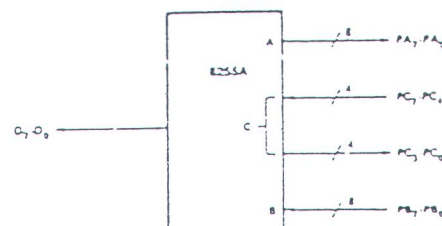
CONTROL WORD #9

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	1	0	0	0	1



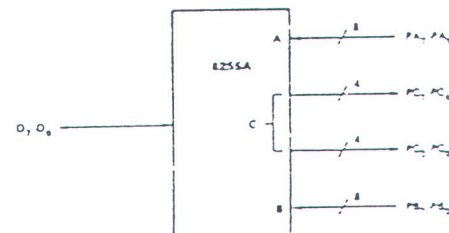
CONTROL WORD #6

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	0	1	0	1	0



CONTROL WORD #10

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	1	0	0	1	0

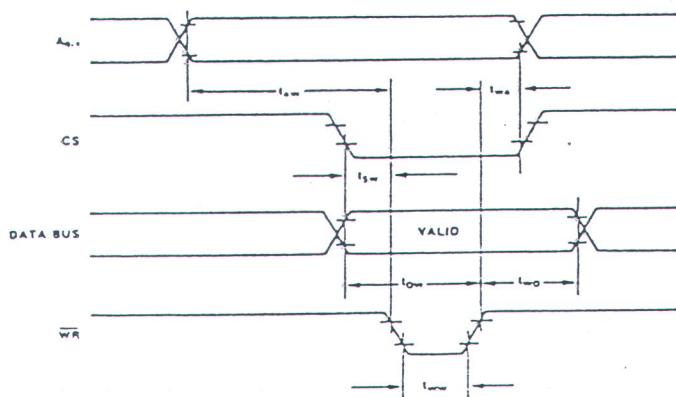


LAMPIRAN

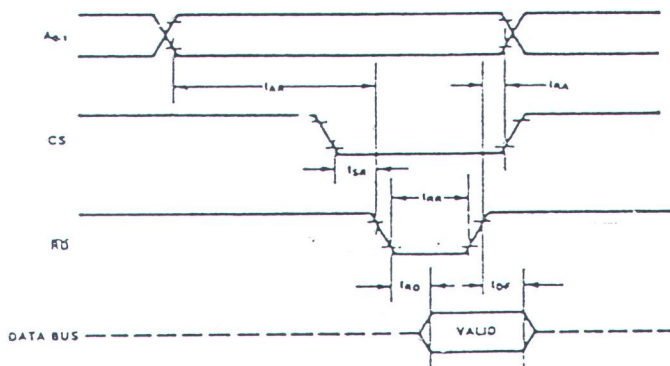
8255A/8255A-5 PROGRAMMABLE PERIPHERAL INTERFACE

WAVEFORMS

WRITE



READ



RECOVERY



RIWAYAT HIDUP

A. Syamsul Arifin, anak ketujuh dari sembilan bersaudara. Dilahirkan di kota Surabaya pada tanggal 20 Februari 1978 dari pasangan H. Usman dan Hj. Siti Maimunah. Bertempat tinggal di jalan Kalimas Baru II Gg. Lebar No. 65 Surabaya - Jawa Timur.

Riwayat pendidikan :

1. Th. 1990 : Lulus SDN Perak Utara IV No.61 Surabaya.
2. Th. 1993 : Lulus SMP Ta' miriyah Surabaya.
3. Th. 1996 : Lulus SMA Ta' miriyah Surabaya.
4. Th. 1996 : Diterima di D III Teknik Elektro Bidang Studi Komputer Kontrol
– Fakultas Teknologi Industri – ITS Surabaya .

Pengalaman Organisasi,

Organisasi dan kegiatan akademik yang pernah diikuti selama menjadi mahasiswa D III Teknik Elektro Bidang Studi Komputer Kontrol – Fakultas Teknologi Industri - ITS :

1. Menjadi anggota Himpunan Mahasiswa Computer Control (HMCC) FTI –

RIWAYAT HIDUP

Sam Hadi Hidayat, anak ketiga dari lima bersaudara. Dilahirkan di kota Samarinda pada tanggal 30 Oktober 1978 dari pasangan H. Idrus Arsani, SH dan Hj. Salbiah. Bertempat tinggal di jalan Cermay No. 7 Samarinda.

Riwayat pendidikan :

1. Th. 1990 : Lulus SDN 79 Samarinda.
2. Th. 1993 : Lulus SMPN 6 Samarinda.
3. Th. 1996 : Lulus SMA Cokroaminoto Samarinda.
4. Th. 1996 : Diterima di D III Teknik Elektro Bidang Studi Komputer Kontrol –
Fakultas Teknologi Industri – ITS Surabaya.

Pengalaman Organisasi,

Organisasi dan kegiatan akademik yang pernah diikuti selama menjadi mahasiswa D III Teknik Elektro Bidang Studi Komputer Kontrol – Fakultas Teknologi Industri - ITS :